Imperial College London

Imperial College of Science, Technology and Medicine

Department of Computing

MSc Advanced Computing

Causal Discovery from Nonstationary Data With Independent Changes: Relaxing Causal Consistency

Author:
Oliver Kanders

CID:
06019970

Imperial College Supervisor:
Wayne Luk

February 17, 2025

Abstract

In many applications, the assumption that causal relationships remain constant is untenable, as underlying mechanisms may change over time or across different domains, limiting the utility of many causal discovery models. Our work extends the constraint-based causal discovery from nonstationary data framework of Constraint-based causal Discovery from heterogeneous/NOnstationary Data (CD-NOD), as we relax the assumption of consistent causal direction (our main challenge, and thus referred to as Challenge 1). To relax the causal consistency assumption within CD-NOD (Challenge 1), we must first survey constraint-based algorithms in nonstationary contexts, using the CD-NOD method as the foundation of our approach by examining its significant contributions: from recovering the causal skeleton of observed variables to determining causal orientations based on independent changes in the data distribution. Our objective is to then explain how our method re-assesses these orientations and to attempt to identify if a potential direction change occurred over the course of time. We resolve Challenge 1 through our algorithm that makes use of a kernel-based local linear regression to estimate the local causal effect and detect sign flips in the estimated coefficients, serving as indicators for changes in causal orientation. Our results showed some promising results in a four node simulation, where our extended algorithm averaged a higher predicted accuracy of recovering causal direction in the presence of a direction reversal than the original CD-NOD algorithm. We hope this work opens the space for further research into integrating the relaxation of causal consistency at the direction orientation processing level.

Contents

1 Introduction

The field of causal discovery remains an important area of interest as machine learning methods have emerged ubiquitous over the past few years. Much of the basis of these approaches relies not only on the assumption that the provided data is i.i.d. but also that effective predictions can be mounted on correlative activity [10, 13]. However, what if distributions shift and spurious correlations come about [17], or notably, that correlation is not the same as causation.

The identification of a cause and its effect is a necessary step for anyone who wants to take a correct measurement of an action and its explicit consequence. The causal framework can be decomposed into phases of causal discovery, causal inference, and causal explainability [17]. Causal discovery, the focus of this report, attempts to derive and determine causal relationships from the data through independence tests [17]. From this causal network, inferences and explanations can be made.

Causal inference fundamentally depends on a formal description of interactions among observed variables, typically via a causal graph. The graphical representation as the storage medium allows for highly effective explainability. The construction of such a graph involves representing a cause-to-effect (outcome) relationship by drawing arrows from the cause to the effect, capturing a qualitative depiction of the system that we wish to investigate [17]. The causal models can uncover deeper understandings of relationships between variables, or nodes on the graph, in contrast to contemporary machine learning frameworks, that are potentially forming a prediction from a black-box model. When the causal graph itself is unknown, it can be inferred by combining available data with prior knowledge. This procedure, referred to as causal discovery, attempts to learn graphical structures that admit a causal interpretation [17, 12].

We will be applying causal discovery in a time series context, a setting in which nonstationarity and temporal ordering arises in fields such as finance and the earth sciences [3, 11, 12]. Most causal discovery methods can be categorized into three classes: constraint-based [15, 7], score-based [1], and functional causal model algorithms. We will be exploring constraint-based algorithms further.

Constraint-based causal discovery algorithms determine causal dependency between nodes by subjecting each graphical edge to statistical independence tests among conditional and marginal probabilities to establish

the network structure [3, 15, 11]. These constraint-based algorithms operate under the assumption of faithfulness, where any conditional independence statement implied by the data distribution is reflected by a corresponding separation in the graph. As faithfulness holds, a graph is said to become a "perfect map" for a probability distribution if and only if the set of conditional independencies implied by the graph is exactly the same as those present in the probability distribution (what is in the data) [3, 15, 17].

A constraint-based algorithm begins by constructing a complete undirected graph, connecting every pair of variables. The algorithm executes a series of statistical conditional independence (CI) tests to check whether variables X and Y are independent when conditioned on a third set of variables Z. If the test suggests that X and Y are independent given Z, the edge linking X and Y in the graph is removed [15, 11]. Because independence tests can become computationally expensive when conditioning on large subsets of variables, limitations are set in regard to the size of the conditioning sets. After the set of relevant edges has been identified, a second phase follows orientation rules to determine the direction of the remaining edges. The detection of colliders (often called v-structures) becomes a point of importance, where two incoming edges meet at a central node, and then proceed to orient other edges to avoid introducing cycles [4, 17]. The result is a Completed Partially Directed Acyclic Graph (CPDAG) that describes the equivalence class of all graphs consistent with the observed set of conditional independence statements [17].

There are two notable algorithms in this space. The PC (Peter–Clark) algorithm, which CD-NOD and our work deploys, defines steps to remove edges that fail CI tests, then orienting those edges that remain [15]. Many practical implementations apply heuristic strategies to reduce the total number of tests needed, acknowledging the decreased statistical power that occurs when conditioning on large sets of variables. A major limitation of basic constraint-based methods is their assumption of no latent confounders and no selection bias. Another algorithm, Fast Causal Inference (FCI)[3, 15], addresses this shortcoming by relaxing those assumptions. It introduces the concept of discriminating paths to determine whether a node in a path is acting as a collider or non-collider, even in the presence of latent variables or biased sampling.

Our work upholds several standard assumptions from causal network literature. We assume pseudo causal sufficiency, meaning that any potential confounders can be modeled as a smooth function of time; thus, the time-

indexed node effectively represents all such confounders [6, 12]. We also assume causal faithfulness, which claims that the conditional independencies observed in the data accurately reflect the absence of direct causal links between variables [6],[12]. Lastly, we assume randomness, so that data points are randomly sampled from the population defined by the causal model, eliminating any selection bias [6, 12].

This report will review three prominent phases outlined by CD-NOD, and then offer our contribution that hopes to resolve partially one of the pervading questions from CD-NOD: "What if some causal directions also change across domains or over time?"[6] (p.46). This question perfectly captures Challenge 1 and will serve as useful point of reflection as we think about the function of each phase in CD-NOD, and potential areas of improvement. CD-NOD included the causal consistency assumption in its framework, and its inability to recognize causal direction switches becomes quite clear as we note its frequent use of invariance principles. CD-NOD can be decomposed across the following phases. Phase 1 recovers the causal graph skeleton by CPC. Phase 2 infers causal directions by generalization of invariance. Phase 3 infers the causal directions between two connected variables whose causal modules are both nonstationary [7, 6, 12]. Our contribution of Phase 4 relaxes the assumption of causal direction consistency, and constructs a detection method for data that may experience a direction flip over the course of the time series.

After we explain the prerequisite framework of the CD-NOD, we will address Challenge 1 and demonstrate how our extended CD-NOD approach detects when a causal coefficient $\hat{\beta}(t)$ changes sign and how it reorients edges accordingly. Our results align with our hypothesis that embedding causal modules into a continuous kernel space (combined with a local linear regression to track $\hat{\beta}(t)$ and detect sign flips) enables the extended CD-NOD method to accurately identify dynamic causal reversals. For example, our four-node simulation (V_1, V_2, V_3, V_4) effectively isolates the edges undergoing reversal. In the following literature review, we survey existing work on causal discovery in time series and nonstationary environments, situating our contribution within the broader context of CD-NOD and constraint-based methods.

2 Literature Survey

Considering that this work stands as an advancement of the work of Huang et al. 2020 [6], we are primarily motivated to have an in depth understanding of the strategies that they lay out. We will also identify potential areas that require further attention in our overall mission to resolve Challenge 1.

2.1 Causal Discovery from Heterogeneous/Nonstationary Data with Independent Changes

2.1.1 Phase 1: Changing Modules Detection & Recovery of Causal Skeleton

In this phase, the classical CD-NOD approach focuses on two necessary assumptions and a flexible structural equation model (SEM) to both detect variables with changing causal modules and recover the underlying causal skeleton among the observed variables. This constraint-based approach takes advantage of the notion of pseudo causal sufficiency, which allows us to use a known variable C (typically representing a domain or time index) as a surrogate for otherwise unobserved, domain or time-dependent confounders.

Assumptions

Assumption 1 (Pseudo Causal Sufficiency): Causal sufficiency is the assumption that all common causes (confounders) of the observed variables are themselves observed. In our framework, unobserved confounders—referred to as pseudo confounders—are assumed to be deterministic (or smooth) functions of a known variable C. Because C is observed (as a domain index or time index), these confounders do not behave as arbitrary latent factors. Rather, they change in a structured way with C, meaning that within any given domain or time instance, their effects are fixed. This relaxation of the full causal sufficiency assumption allows us to detect changes in causal mechanisms by testing the dependence between each variable and C [6, 12].

Assumption 2 (Markov and Faithfulness Assumption): The faithfulness assumption claims that all conditional independence relationships present in the joint distribution correspond exactly to the d-separation relations in the causal graph [3, 17]. Our joint distribution over the observed variables V and the functions of C (which capture both the pseudo confounders and the changing parameters) is assumed to be Markov and faithful

to an augmented causal graph G_{aug}. Therefore, any conditional independence observed in the data truly reflects the absence of a direct causal connection [3, 6, 16].

Structural Equation Model (SEM)

Phase 1 is based on a flexible SEM that allows the causal mechanisms to vary with C [6, 7, 4, 17, 16]. For each observed variable V_i, the SEM is given by

$$V_i = f_i(\text{PA}_i, g_i(C), \theta_i(C), \epsilon_i), \tag{1}$$

where: PA_i denotes the set of direct causes (parents) of V_i, $g_i(C)$ represents the pseudo confounders affecting V_i, which are functions of the observable variable C, $\theta_i(C)$ are parameters of the causal mechanism that can also change with C, ϵ_i is an independent noise term ensuring that the model is stochastic. This formulation does not impose any strict parametric form, and is able to model the complex, nonlinear relationships in the data [6, 7].

Detection of Changing Modules

Under these assumptions and based on the SEM model, we are able to determine if a variable's causal mechanism is invariant or changing with C. This is done by asking: *Is the variable V_i conditionally independent of C given a set of other variables* [6]? If V_i is found to be conditionally independent of C, this implies that its underlying causal module remains constant across domains or time. If no such conditional independence is found, it suggests that the causal mechanism for V_i varies with C [6, 7]. By including C in a constraint-based skeleton-learning algorithm (similar to the PC algorithm), we can simultaneously identify which variables exhibit domain or time-dependent changes and reconstruct the causal skeleton of the graph. The learning of the causal skeleton is the first step in calculating the ultimate direction of our graph. We must understand the dependency structure of the graph by uncovering the skeleton, which shows the connections of each V_i and C as undirected relationships. Phase 2 will investigate further how orientation can be gathered from these edges.

The algorithm initializes a complete undirected graph U_G over the set $V \cup \{C\}$, where $V = \{V_1, V_2, \ldots, V_n\}$ is a collection of observed variables and

Algorithm 1 Phase 1: Causal Skeleton Recovery via CPC

Input: Variable set $V = \{V_1, V_2, \ldots, V_n\}$ and context variable C
Output: Undirected graph U_G

$U_g \leftarrow \text{CompleteGraph}(V \cup \{C\})$

for each $V_i \in V$ **do**
    for each subset $S \subseteq \{V_k : k \neq i\}$ **do**
        if IndependenceTest$(V_i, C \mid S)$ is true **then**
            Remove edge between V_i and C in U_g
            break ▷ exit loop over S once independence is detected
        end if
    end for
end for

for each pair of distinct variables (V_i, V_j) in V **do**
    for each subset $S \subseteq \Big(\{V_k : k \neq i \text{ and } k \neq j\} \cup \{C\}\Big)$ **do**
        if IndependenceTest$(V_i, V_j \mid S)$ is true **then**
            Remove edge between V_i and V_j in U_G
            break ▷ exit loop over S once independence is detected
        end if
    end for
end for
return *skeleton* $\leftarrow U_G$

C represents the context variable. All possible dependencies among the variables are considered prior to any pruning based on conditional independence tests. For each $V_i \in V$, we evaluate the conditional independence between V_i and C by conditioning on all subsets $S \subseteq \{V_k : k \neq i\}$. For each such S, we test $V_i \perp C \mid S$. If there exists a subset S such that the conditional distribution $P(V_i \mid S)$ is invariant with respect to variations in C, then the causal module/variable responsible for generating V_i is independent of C (note: we still do need to account for the other variables within S). Under this invariance, C does not contribute any additional information about V_i beyond what is already captured in S. In this case, the edge connecting V_i and C in U_G is removed. However, if no subset S satisfies $V_i \perp C \mid S$, we conclude that the mechanism generating V_i is nonstationary with respect to C, and the edge between V_i and C holds.

The last step is repeating this same process, but now testing for marginal and conditional independence among the observed variables V. For every pair of distinct variables $V_i, V_j \in V$ (with $i \neq j$), we look at conditioning sets of the form $S \subseteq (\{V_k : k \neq i, j\} \cup \{C\})$ and test whether $V_i \perp V_j \mid S$. If there exists any such conditioning set S for which the conditional independence holds, the edge between V_i and V_j is removed from U_G.

2.1.2 Phase 2: Infer Causal Directions by Generalization of Invariance

After Phase 1 identifies the causal skeleton (which pairs of variables are connected by an edge), Phase 2 tracks the distribution shifts captured in C to orient certain edges. The power of C emerges as when causal modules change across domains or time, those changes can reveal the true direction of causation.

Two important concepts arrive here: C-Specific Variables and their Unshielded Triples, and the Generalization of Invariance [6]. A variable V_k is called C-specific if it was found in Phase 1 to have a changing causal mechanism with respect to C, observing that V_k is adjacent to C. Then, for each variable V_i that is connected to V_k in the newly learned skeleton, a so-called triple $(C - V_k - V_i)$ is inspected. If C and V_i are not directly connected, this triple becomes what is known as an unshielded triple, and orientation rules can assist in resolve the direction of the edge to be $V_k \to V_i$ or vice versa $V_k \leftarrow V_i$ [15, 6].

Complementing this is the principle of invariance [10]. The principle

states that if the conditional distribution $P(V_i \mid S)$ remains stable across values of C, then V_i does not directly depend on C via S. We are observing $P(V_i \mid V_k, C)$ to verify stability as C varies. When this conditional distribution is invariant, any influence C might have on V_i is fully mediated by V_k ($V_k \to V_i$) [6]. Conversely, if $P(V_i \mid V_k, C)$ varies with C while the marginal distribution $P(V_k)$ is stable across different values of C, V_k will not be directly influenced by C. The variability in $P(V_i \mid V_k, C)$ may suggest that the changes in V_i are not completely explained by V_k alone, and that V_i is the driving causal module $V_k \leftarrow V_i$. Testing these invariance properties supports the determination of causal direction within an unshielded triple, expressing the difference of whether V_i is the parent or child relative to V_k [6].

Algorithm 2 Phase 2: Causal Direction Identification by Generalization of Invariance

Require: U_g: Adjacency matrix from Phase 1, C: Domain/time variable, V: Set of all variables (excluding C).

Ensure: Partially oriented graph U_g.

1: **Step 1:** For each $X \in V$ with $U_g(C, X) \neq 0$, set $C \to X$.

2: **Step 2:** For each unshielded triple $C \to V_k - V_i$ (with $V_i \notin \text{Adj}(C)$):

- If there exists $S \subseteq V \setminus \{V_k\}$ such that $V_i \perp\!\!\!\perp C$, orient $V_k \leftarrow V_i$.
- Otherwise, if independence holds only when V_k is included, orient $V_k \to V_i$.

3: **Step 3:** Output the updated U_g.

In Phase 2, we use the generalization of invariance to orient the edges in the causal skeleton identified during Phase 1. We start with an unshielded triple ($C \to V_k - V_i$). For Step 2, when C is adjacent to V_k but not V_i, test if V_i becomes independent of C with or without V_k in the conditioning set. If V_i is independent of C only when V_k is excluded, then orient $V_k \leftarrow V_i$; otherwise, orient $V_k \to V_i$. Algorithm 2 does not observe the event that C is adjacent to both V_k and V_i.

2.1.3 Phase 3: Inferring Causal Directions by Independent Change Principle

The greatest contribution of Huang et al. 2020 appears in Phase 3, which constructs an alternative algorithm to observe and measure the dependence between changing modules [6, 7]. Phase 2 was based on the idea that one of the causal modules is invariant, while the other changes ($P(V_i)$ is invariant but $P(V_j \mid V_i)$ changes). However, in a nonstationary environment, it is more likely that the marginal distribution $P(cause)$ and the conditional distribution $P(effect \mid cause)$ vary across changing values of C. Phase 3 addresses this concept of when causal modules (the marginal distribution and the conditional distribution) change independently. This nonstationarity allows for an additional layer of inference for the causal direction.

Causal Direction Identification by Independently Changing Modules In a nonstationary environment, the distributions associated with each variable may change over time or across domains. If one variable causes another, then the way its marginal distribution changes (the causal module $P(\text{cause})$) should be independent of how the conditional distribution of the effect given the cause changes ($P(\text{effect} \mid \text{cause})$) [7, 6]. For example, consider two adjacent variables V_1 and V_2 that are also both adjacent to a context variable C. If the true causal direction is $V_1 \to V_2$, then the variations in $P(V_1)$ (which depend on some parameter $\theta_1(C)$) should be independent of the variations in $P(V_2 \mid V_1)$ (which depend on $\theta_2(C)$). In the reverse (incorrect) direction, however, the effective parameters in the reversed decomposition would be jointly determined by θ_1 and θ_2, leading to dependent changes [6].

Kernel Embedding of Constructed Joint Distributions: Directly embedding a conditional distribution $P(Y \mid X, C = c)$ is challenging. Instead, the method constructs a joint distribution:

$$\widetilde{P}(V_j, V_i \mid C = c) = P(V_j \mid V_i, C = c)\, P(V_i) \tag{2}$$

possibly scaled by $P(C = c)$ [6]. Equation 2 is able to separate the conditional distribution ($P(V_j \mid V_i, C = c)$) from the marginal distribution ($P(V_i)$). Because $P(V_i)$ is assumed invariant or constant, changes in $P(V_j \mid V_i, C = c)$ are reflected in $\widetilde{P}(V_j, V_i \mid C = c)$. Equation 2 isolates the causal mechanism and becomes fundamental to Phase 3, as independent changes in the causal module can be detected if $P(V_i)$ is stable while $P(V_j \mid V_i, C = c)$ varies with c. Kernel embedding allows us to represent these distributions

as elements in a Reproducing Kernel Hilbert Space (RKHS) [7, 6]. With the kernel trick, one can compute the corresponding Gram matrices (using either linear or Gaussian kernels) over the entire dataset without resorting to window segmentation [6]. These Gram matrices are then used to compute a normalized version of the Hilbert–Schmidt Independence Criterion (HSIC), which measures the dependence between the kernel embeddings of the causal modules. For the two-variable case, if the dependence between the embeddings of $P(\text{cause})$ and $P(\text{effect} \mid \text{cause})$ is lower than that for the reverse decomposition ($P(\text{effect}) \cdot P(\text{cause} \mid \text{effect})$), the causal direction is inferred to be from the cause to the effect.

Causal Direction Determination in the Multi-Variable Case

When more than two variables are involved, potential confounders—variables that affect both V_l and V_k—can lead to misleading causal inferences if not properly controlled. Phase 3 answers this by introducing *deconfounding sets*, which block all paths that could otherwise create or mask dependencies between V_l and V_k.

Deconfounding Sets For each unoriented edge (V_l, V_k) in the partially oriented graph, two sets of variables are defined [6]:

Minimal Deconfounding Set $Z_{lk}^{(1)}$ [6]: This is the smallest set of variables (all adjacent to V_l) such that:

1. None of them is a descendant of V_l or V_k.
2. Conditioning on $Z_{lk}^{(1)}$ blocks every path into V_l or V_k that has an arrow into these nodes.
3. If any variable is removed from $Z_{lk}^{(1)}$, the set no longer blocks all such paths (i.e., it is minimal).

Minimal Potential Deconfounding Set $Z_{lk}^{(2)}$ [6]: This set also consists of variables adjacent to V_l, with the same descendant restriction, but now blocks every path *not containing an arrow out of* V_l or V_k. Minimality again requires that removing any variable breaks this blocking property.

The idea is that $Z_{lk}^{(1)}$ and $Z_{lk}^{(2)}$ collectively account for different types of confounding paths that can distort the causal relationship between V_l and V_k.

Iterative Dependence Testing Once $Z_{lk}^{(1)}$ and $Z_{lk}^{(2)}$ are identified, iterative conditional tests begin [6].

1. **Select Subsets of the Potential Deconfounding Set** Let n be an integer that determines the size of the conditioning set. The algorithm picks a subset $Z_{lk}^{(2,n)} \subseteq Z_{lk}^{(2)}$ so that

$$|Z_{lk}^{(1)}| + |Z_{lk}^{(2,n)}| = n.$$

   By incrementing n step by step, the algorithm works on progressively larger (or different) conditioning sets.

2. **Compute Kernel Embeddings** For each unoriented pair (V_l, V_k), the algorithm embeds [6]:

$$P\big(V_k, Z_{lk}^{(1)}, Z_{lk}^{(2,n)}\big) \quad \text{and} \quad P\big(V_l \mid V_k, Z_{lk}^{(1)}, Z_{lk}^{(2,n)}\big)$$

   into a Reproducing Kernel Hilbert Space (RKHS). It then uses the Hilbert–Schmidt Independence Criterion (HSIC) to measure how dependent these two distributions are.

3. **Compare the Reverse Direction** The same procedure is applied to [6]:

$$P\big(V_l, Z_{lk}^{(1)}, Z_{lk}^{(2,n)}\big) \quad \text{and} \quad P\big(V_k \mid V_l, Z_{lk}^{(1)}, Z_{lk}^{(2,n)}\big),$$

   allowing the algorithm to test which orientation ($V_l \to V_k$ or $V_k \to V_l$) shows lower dependence.

4. **Orient the Edge** If one direction consistently demonstrates lower dependence between the marginal and conditional modules, that direction is taken as the causal arrow. If ambiguity still occurs after the iterations (perhaps in the presence of pseudo confounders), standard propagation rules (Meek's rules) can be deployed to construct final orientations [6, 9].

By iteratively selecting and testing these deconfounding sets, Phase 3 blocks confounding paths while measuring the (in)dependence of changing causal modules. This allows the method to infer the correct causal directions in multi-variable settings, even when both the marginal and conditional distributions vary across different contexts.

3 Experimentation

3.1 Benchmarks

We re-created many of the synthetic data benchmarks from Huang et al. 2020 [6], allowing us to then compare how our model with the extension of Phase 4 evaluates against the original.

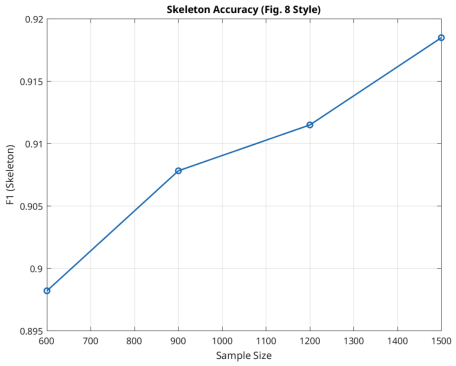
3.1.1 Settings 1-3

Setting 1: Runs experiments on randomly generated 6-variable DAGs (with 4–6 changing variables) over various sample sizes. It simulates nonstationary data and evaluates multiple metrics (skeletal, changing-module, direction, and CPDAG F1) to assess overall CD-NOD performance. A CPDAG (Completed Partially Directed Acyclic Graph) represents a Markov equivalence class of DAGs [17], and this CPDAG accuracy calculates F1 by treating undirected edges (i–j) as correct unless they explicitly reverse a directed edge from the ground truth.

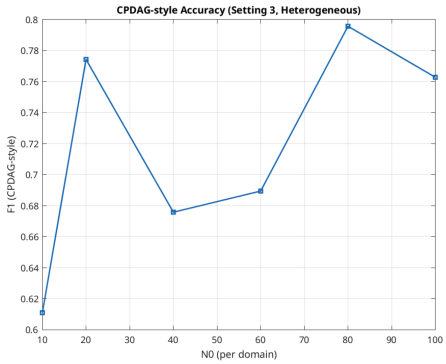
Setting 2: Observes fully connected DAGs (we demonstrate with four nodes) where the skeleton is forced to remain fully connected ($\alpha = 1$). This setting highlights Phase 3 for orientation identification and evaluates directional performance (CPDAG: precision, recall, and F1).

Setting 3: Generates heterogeneous multi-domain data (10 domains with varying per-domain sample sizes) from a random DAG. The setting measures how well CD-NOD recovers skeleton and directed structure (both strict direction and CPDAG F1) across domain shifts.

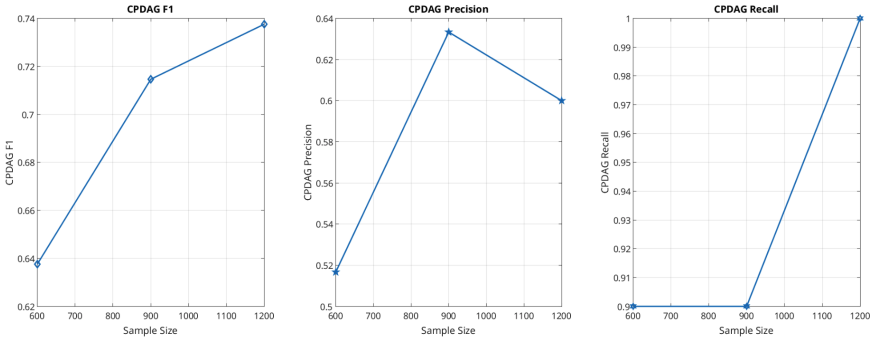
F1 Scores in Settings 1-3: Skeleton, CPDAG



(a) Setting 1



(b) Setting 3



(c) Setting 2

Figure 1: Overview of the three experimental settings: (a) Skeleton accuracy for Setting 1, (b) Multi-domain data in Setting 3, (c) Fully connected DAG in Setting 2.

Additional Results for Setting 1:

Sample Size	F1 (mean±std)	Precision	Recall
600	0.963±0.074	0.976	0.958
900	0.970±0.056	0.986	0.960
1200	0.979±0.054	0.976	0.986
1500	0.978±0.048	0.984	0.976

Table 1: Nonstationary Data: Changing-Module Accuracy for Setting 1.

Both Figure 1 and Table 1 display similar results to Huang et al. 2020 [6]. We would like to note that our model was tuned differently to that of Huang et al. 2020, and so exact replication was a difficult mark to reach.

3.2 Challenge Resolution: Phase 4 (Direction Identification)

We return to our challenge that causal consistency may be a limiting assumption. Many real-world systems exhibit precisely the challenge that causal directions themselves may evolve. A causal influence that was valid in one time regime could reverse direction in another. For example, the sign of a causal effect might change from positive to negative over a long time horizon, or due to changing environmental conditions. In financial markets, an instance could be a lead-lag relationship switching when a commodity transitions from a demand-driven regime to a supply-driven one [12].

In the prior three phases, we have recovered the causal skeleton (Phase 1), inferred partial orientation (Phase 2), and then deployed the independent change principle when both the marginal and conditional distributions were changing with respect to C (Phase 3). These steps, however, do not specifically address the setting of a causal relationship changing across time: once we had identified a pair $X \to Y$ (or $X \leftarrow Y$) in earlier phases, that orientation was assumed consistent throughout the entire dataset. Now, we propose a mechanism to review all the orientations from phases 2/3, and isolate potential reversals. Phase 4 estimates a *local causal coefficient* over time or domain indices using a kernel-based regression approach, assigning higher weight to observations that are closer in the context variable C. This local regression produces a continuous trajectory of slope estimates that reflect how the functional effect of a cause on its effect develops with C [5, 2]. If we observe a sign change of sufficient magnitude in these local slopes, this

can serve as a promising indicator that in some regions of C, the variable is acting as a cause (increasing the effect), while in other regions it might be acting more like an effect (the influence is reversed)[2].

Function	**Concept from CD-NOD** [6]	**Section** [6]
`continuousCoefficientFunction()`	Kernel-based estimation	Section 4.2.1
`hasSignFlip()`	Independent change detection	Section 4.2.2
`findEdgesWithCode()`	Identifying flipped edges	Algorithm 3
`applyMeekRulesIgnoringFlips()`	Refining causal graph	Section 4.3

Table 2: Phase 4 aims to determine direction orientation based on the kernel embedding approach from Phase 3.

3.2.1 The Kernel Embedding Basis

This approach draws inspiration from Phase 3, which introduces the concept of kernel embeddings for capturing smoothly varying causal modules as C changes. By embedding this joint distribution into a Reproducing Kernel Hilbert Space (RKHS), we can represent the causal module in a continuous manner. The corresponding Gram matrices—computed using linear or Gaussian kernels over the entire dataset—are then used to measure the dependence between the causal modules via a normalized Hilbert–Schmidt Independence Criterion (HSIC) [6]. From these kernel embeddings, we can detect whether the independent changes in the cause and effect modules support one causal direction over the reverse. Rather than discretizing C into separate domains, CD-NOD method treats C as a continuous variable and weights observations by their proximity to each point c_n. Similarly, `continuousCoefficientFunction` applies Gaussian kernel weighting to estimate local linear models at each t_0, tracking how the causal influence between V_i and V_j changes continuously over C.

3.2.2 Local Linear Regression with Gaussian Kernel Weighting

In nonstationary settings, the causal module $P(V_j \mid V_i, C = c)$ is expected to vary smoothly with the continuous context variable C. To estimate the behavior of this module at a context value t_0, we assign weights to each observation based on how the proximity of the context c_i is to t_0. This

localized weighting is achieved using a Gaussian kernel [14], defined as

$$w_i(t_0) = \exp\left(-\frac{(c_i - t_0)^2}{2\,\lambda^2}\right), \tag{3}$$

where c_i is the context (e.g., time or domain) associated with the ith observation, and λ is the bandwidth parameter controlling the decay rate. This function originates from the standard Gaussian density [14]

$$K(x, x') = \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{||x - x'||^2}{2}\Big),$$

but with the normalization constant omitted since we are interested in relative weights. The exponential decay guarantees that observations closer to t_0 are given much higher weight than those further away [14]. This procedure constrains the estimation to a local neighborhood around t_0, helping generate an accurate assessment of the causal module's behavior at that point.

Once the Gaussian weights are computed, we perform a local linear regression at each target context t_0 to estimate how the causal relationship between V_i (predictor) and V_j (response) changes. The regression model is fitted using weighted least squares [5]:

$$\hat{\beta}(t_0) \;=\; \Big(\mathbf{M}^T\, W(t_0)\, \mathbf{M}\Big)^{-1} \Big(\mathbf{M}^T\, W(t_0)\, \mathbf{y}\Big), \tag{4}$$

where

$$\mathbf{M} = \begin{pmatrix} 1 & V_i(1) \\ 1 & V_i(2) \\ \vdots & \vdots \\ 1 & V_i(T) \end{pmatrix}, \quad W(t_0) = \operatorname{diag}\Big(w_1(t_0),\, w_2(t_0),\, \ldots,\, w_T(t_0)\Big),$$

and

$$\mathbf{y} = \begin{pmatrix} V_j(1) \\ V_j(2) \\ \vdots \\ V_j(T) \end{pmatrix}.$$

This regression provides an estimate of the local slope $\hat{\beta}(t_0)$, which serves as an indicator of the strength and direction of the causal effect at the context t_0 [5].

3.2.3 Flip Detection

A flip in the sign of $\hat{\beta}(t_0)$ demonstrates a notable shift in the causal module. The vector of slope estimates $\{\hat{\beta}(t_0)\}$ over a grid effectively traces the trajectory of the embedded causal module, allowing us to detect nonstationary behavior.

For instance, if the true causal effect transitions from positive to negative at some context c, then there will be a sign flip in $\hat{\beta}(t_0)$ as computed in Equation (4). We formalize the detection of a sign flip by the criterion [14]:

$$\min_{t_0} \hat{\beta}(t_0) \cdot \max_{t_0} \hat{\beta}(t_0) < 0 \quad \text{and} \quad \max_{t_0} \hat{\beta}(t_0) - \min_{t_0} \hat{\beta}(t_0) > \text{threshold}\,. \tag{5}$$

This condition ensures that the estimated slope changes sign significantly, serving as evidence for a nonstationary change in the causal module.

3.2.4 Results

We ran an experiment, using the same Gaussian-Process based nonstationary data generation technique as in Huang et al 2020 [6]. In this technique, each edge $i \to j$ of the DAG is assigned a time varying coefficient $b_{ij}(t)$ modeled by this Gaussian-Process [6]. After assigning random Gaussian Process curves to each existing edge, we modify these coefficients to create two distinct regimes (e.g. the edge $1 \to 3$ and then $3 \to 1$). Over the first section of the time series, $b_{1,3}(t)$ is nonzero and $b_{3,3}(t) = 0$. This creates the edge direction $1 \to 3$. As t passes into the next regime $t > T/2$, or any chosen break (we went with naive case of two regimes), this relationship reverses and we set $b_{1,3}(t) = 0$ and make $b_{3,1}(t)$ nonzero, creating the edge direction $3 \to 1$. This two regime flip occurs in reality all the time. Consider a commodity such as oil, where in a demand-driven regime, oil prices may lead gasoline prices $(1 \to 3)$. However, in a supply-driven regime, the market shifts and direction reverses $(3 \to 1)$. The following experiments are based on this kind of data generation.

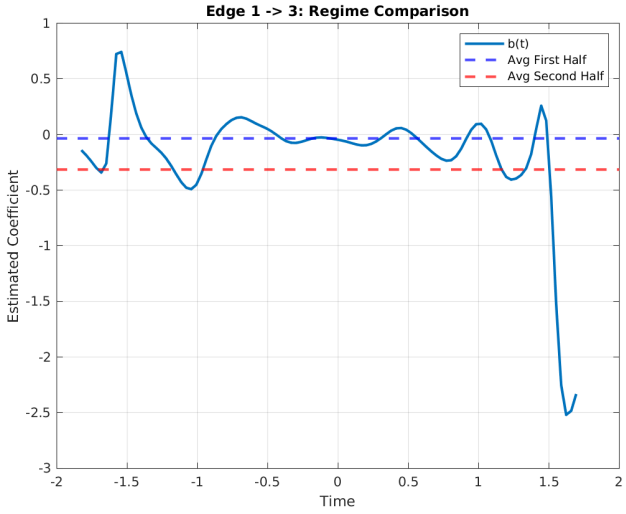
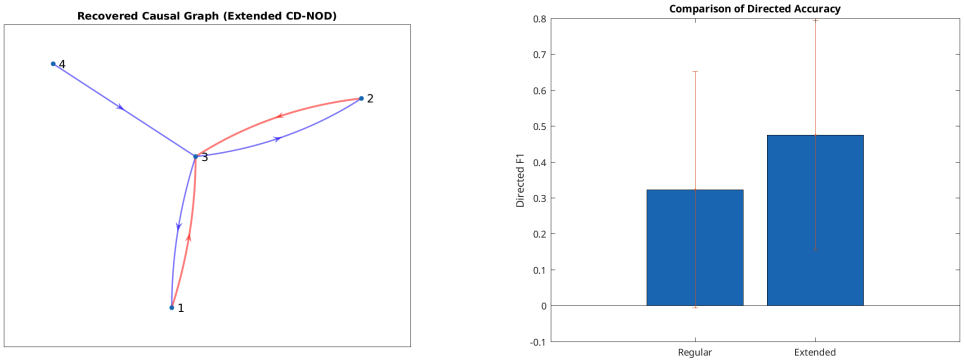


Figure 2: Transition of the function $\hat{\beta}(t)$ over the context variable t



(a) Recovered Flip Graph

(b) Direction Accuracy Comparison

Figure 3: Comparison of the recovered flip graph and direction accuracy.

Figure 2 plots the function $\hat{\beta}(t)$, denoted by $b(t)$, as a function of the time or context variable t. Visually, the curve crosses the zero line multiple times, suggesting that $\hat{\beta}(t)$ transitions between positive and negative values. Equation (5) constitutes a true direction flip if the following two conditions hold: $\min_{t_0} \hat{\beta}(t_0) \cdot \max_{t_0} \hat{\beta}(t_0) < 0$, $\max_{t_0} \hat{\beta}(t_0) - \min_{t_0} \hat{\beta}(t_0) >$ threshold.

Figure 2 demonstrates multiple crossings of the x-axis, indicating many negative product instances and satisfying the first condition. However, we need to establish a sufficiently large range, which is only recognized around $t_{1.5}$: a direction flip occurs in the second half of the regime (delayed a bit with overlap of the previous regime), it becomes more defined as we get closer to $t \to T$.

Figure 3a, the recovered causal graph (using the extended CD-NOD algorithm) shows four observed variables labeled V_1, V_2, V_3, and V_4. V_3 lies at the center, receiving a direction arrow from V_4 and sending direction arrows to V_1 and V_1. This structure suggests that V_4 is a parent of V_3 ($Pa(V_3)$), and V_3 induces an effect on V_1 and V_2. As described in Phase 4, the extended algorithm checks for nonstationary flips in causal relationships over time (or over a continuous context) and reorients edges if it detects a significant change in $\hat{\beta}(t)$. In this trivial case of four nodes, with at most one flip allotted per changing module, the final graph displays a corrected set of directions. V_4 exerts causal influence on V_3, and nodes V_1 and V_2, having been exposed to a sign flip, exert a causal influence on V_3 (the red arrows describe the flipped direction).

Figure 3b compares the directed F1 score of the regular CD-NOD algorithm (left bar) with that of the extended CD-NOD algorithm (right bar). Directed F1 measures how accurately each method recovers the true causal directions, especially in the latter portion of the data where some edges may reverse. Directed F1 measures how accurately the predicted adjacency matrix compares to the true directed adjacency matrix: this measurement focuses on how many oriented edges were discovered correctly in the data. The extended approach proves valuable in detecting causal coefficient changes ($\hat{\beta}(t)$) in sign and magnitude. In a nonstationary environment where causal direction fluctuates, the extended method consistently achieves a higher mean directed F1 than the standard version (≈ 0.33 vs. ≈ 0.47), with error bars (representing standard deviations across multiple simulations) further establishing that even with variability the extended CD-NOD method outperforms the regular algorithm in recovering correct edge orientations.

—

4 Conclusions

We hoped to extend the classic CD-NOD algorithm with a design that has the capacity to address causal direction flips within nonstationary settings (Challenge 1). The method employs a kernel-based approach to estimate local causal coefficients and to detect direction flips in the causal relationships as the context changes. Experimental results on synthetic data demonstrate that the extended algorithm recovers the true causal directions more accurately than the standard CD-NOD method in the presence of a varying causal relationships over time. Our extension returned higher directed F1 scores compared to the standard approach, displaying around a 40% improvement when true causal directions reverse over time.

Although Phase 4 serves as an effective corrective step when the CD-NOD algorithm attempts to model causally inconsistent data (where causal direction flips), standard causal discovery frameworks assume that causal directions remain consistent across different contexts. We relaxed this assumption to accommodate the possibility that causal directions may change over time, and to build a phase that could extend the applications of CD-NOD. There still lies a fundamental concern that treating the relaxation of causal consistency as a post-processing step would be less effective than integrating it directly at the edge orientation level. If errors occur during Phase 3—such as inaccuracies in estimating local coefficients—then the subsequent reorientation in Phase 4 may not fully correct these errors, leading to a compromised final causal graph. In the future, we hope to construct methods that integrate the relaxation of causal consistency directly into the orientation process, so that the propagation of errors from earlier phases can be prevented

References

[1] Chickering DM. Optimal structure identification with greedy search. *Journal of Machine Learning Research*. 2003 Mar;3:507–554. doi:10.1162/153244303321897717. https://www.jmlr.org/papers/volume3/chickering02b/chickering02b.pdf

[2] Ghassami AE, Kiyavash N, Huang B, Zhang K. Multi-domain causal structure learning in linear systems. *Advances in Neural Information Processing Systems*; 2018. https://proceedings.neurips.cc/paper_files/paper/2018/file/6ad4174eba19ecb5fed17411a34ff5e6-Paper.pdf

[3] Gong C, Zhang C, Yao D, Bi J, Li W, Xu Y. Causal discovery from temporal data: an overview and new perspectives. *ACM Computer Survey* 2024 Dec;57(4):100. doi:10.1145/3705297.

[4] Hasan U, Hossain E, Gani MO. A survey on causal discovery methods for I.I.D. and time series data. *arXiv* [Preprint]. 2024; arXiv:2303.15027. https://arxiv.org/abs/2303.15027.

[5] Hastie T, Tibshirani R, Friedman J. Kernel smoothing methods. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer; 2009. p. 191–216.

[6] Huang B, Zhang K, Zhang J, Ramsey J, Sanchez-Romero R, Glymour C, Schölkopf B. Causal discovery from heterogeneous/nonstationary data with independent changes. *Journal of Machine Learning Research*. 2020;21(89):1–53. https://www.jmlr.org/papers/volume21/19-232/19-232.pdf.

[7] Huang B, Zhang K, Zhang J, Sanchez-Romero R, Glymour C, Schölkopf B. Behind distribution shift: mining driving forces of changes and causal arrows. *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE; 2017. p. 913–918. doi:10.1109/ICDM.2017.114.

[8] Li A, Jaber A, Bareinboim E. Causal discovery from observational and interventional data across multiple environments. In: Oh A, Naumann T, Globerson A, Saenko K, Hardt M,

Levine S, eds. *Advances in Neural Information Processing Systems*. Vol. 36. Curran Associates, Inc; 2023. p. 16942–16956. https://proceedings.neurips.cc/paper_files/paper/2023/file/368cba57d00902c752eaa9e4770bbbbe-Paper-Conference.pdf.

[9] Meek C. Causal inference and causal explanation with background knowledge. *arXiv* [Preprint]. 2013; arXiv:1302.4972. https://arxiv.org/abs/1302.4972.

[10] Perry R, von Kügelgen J, Schölkopf B. Causal discovery in heterogeneous environments under the sparse mechanism shift hypothesis.*Advances in Neural Information Processing Systems*; 2022. https://papers.neurips.cc/paper_files/paper/2022/file/46a126492ea6fb87410e55a58df2e189-Paper-Conference.pdf.

[11] Runge, J., Gerhardus, A., Varando, G. et al. Causal inference for time series. *Nature Reviews Earth and Environment*, 487–505 (2023). https://doi.org/10.1038/s43017-023-00431-y.

[12] Sadeghi A, Gopal A, Fesanghary M. Causal discovery in financial markets: A framework for nonstationary time-series data. *arXiv* [Preprint]. 2024; doi:10.48550/arXiv.2312.17375.

[13] Schölkopf B, Locatello F, Bauer S, Ke NR, Kalchbrenner N, Goyal A, et al. Toward causal representation learning. *Proceedings of the IEEE*. 2021;109:612–630. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9363924.

[14] Schölkopf B, Smola AJ. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press; 2002. https://doi.org/10.7551/mitpress/4175.001.0001.

[15] Spirtes P, Glymour C, Scheines R. *Causation, Prediction, and Search*. 2nd ed. Cambridge, MA: MIT Press; 2001.

[16] Spirtes P, Zhang K. Causal discovery and inference: concepts and recent methodological advances. *Applied Informatics* (Berlin). 2016;3:3. doi:10.1186/s40535-016-0018-x.

[17] Zanga A, Stella F. A survey on causal discovery: theory and practice. *arXiv* [Preprint]. 2023; doi:10.48550/arXiv.2305.10032.

5 Declarations

I acknowledge the use of ChatGPT 4o (OpenAI, https://chat.openai.com/) to check Vancouver reference formatting and for Latex syntax. I confirm that no content generated by AI has been presented as my own work.